

# ナース・スケジューリングにおける 部分問題実行可能解空間のネットワーク表現

秋田 博紀<sup>†</sup>・池上 敦子<sup>†</sup>

(受付 2012 年 12 月 5 日; 改訂 2013 年 4 月 12 日; 採択 4 月 16 日)

## 要 旨

本論文では、ナース・スケジューリングの解空間把握に取り組む。病棟ナースの勤務表作成では、各シフトに適切な人数とスキルレベルのナースを揃えると同時に、それぞれのナースの勤務負荷を考慮しなければならない。ナース・スケジューリングは、組合せ最適化問題として解くことが難しい問題として知られてきたが、これに加え、人間が潜在的に考慮している制約や評価尺度も存在することから、コンピュータや最適化アルゴリズムにとって扱いにくい問題でもある。この問題における意思決定を支援するためには、実行可能解空間を直感的に把握でき、その中の良解の分布に関する情報を提供できるような仕組みが必要である。本論文では、ナース・スケジューリングの部分問題として、1 ナースの最適スケジュールを見つける問題を対象に、この問題の解空間を表現するネットワークを構築する。このネットワークのソースノードからシンクノードまでの全経路は、そのナースの実行可能スケジュールであると同時に、すべての実行可能スケジュールがこのネットワークの経路として含まれる。ある試行解(勤務表)に対し、他のナースのスケジュールを固定した下での最適スケジュールや上位複数の最良スケジュールを、ネットワーク上の最短路や  $k$  最短路を探索することで得られるようになる。本論文では、このネットワークの構築方法と実際に構築した結果を報告する。

キーワード： ナース・スケジューリング, 解空間, 解の可視化, ネットワーク表現, 動的計画.

## 1. はじめに

ナース・スケジューリングは、病棟ナースの各日の勤務を決定する勤務表作成問題である。表 1 に、3 交替制勤務表の例 (Ikegami and Niwa, 2003) を示す。

一般に、1 列目にナース名を記載するが、この例ではナース番号を示してある。1 行目には日付、2 行目には曜日を頭文字で示してある。各行における各日の列に、対象ナースのその日の勤務内容の記号(この例では、一は日勤、e は準夜勤、n は深夜勤、+ はセミナー等のその他の勤務、/ は休み)が示される。そして、各列の下には各シフトの勤務人数、各行の右には対象ナースのその月における休みの回数や各シフトの勤務回数が示されている。

ナース・スケジューリング研究は、1970 年代にアメリカで始まった (Warner, 1976; Miller et al., 1976; Arther and Ravindran, 1981) が、アルゴリズムの適用だけでは思うようなスケジュールを得ることが難しいことから意思決定支援システムの導入が提案された (Bell et al., 1986)。

---

<sup>†</sup> 成蹊大学 理工学部: 〒186-8633 東京都武蔵野市吉祥寺北町 3-3-1

表 1. 3 交替勤務表の例: 3shift-DATA1 (Ikegami and Niwa, 2003)

(一:日勤, e:準夜勤, n:夜勤, +:その他の勤務, /:休み)																																				
Nurse No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		/	-	e	n	+
W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T		/	-	e	n	+	
1	e	e	/	-	-	n	n	/	/	/	-	-	e	/	/	-	-	/	-	-	-	e	e	/	-	e	n	n	/	/	10	10	6	4	0	
2	n	/	/	/	-	-	-	/	e	e	/	+	-	-	n	n	/	/	-	-	-	-	e	e	/	/	-	n	n	/	/	9	11	4	5	1
3	-	/	-	-	/	/	-	e	e	n	n	/	/	-	-	e	n	n	/	/	/	/	-	-	-	e	/	-	-	9	13	4	4	0		
4	/	-	-	e	/	-	e	n	n	/	/	/	-	e	/	/	-	e	/	+	-	-	n	n	/	/	-	-	-	10	11	4	4	1		
5	-	n	n	/	e	e	/	-	-	-	n	n	/	e	e	/	-	-	n	n	/	/	/	/	-	-	/	/	e	10	9	5	6	0		
6	/	-	e	n	n	/	/	/	-	-	/	-	n	n	/	/	-	e	e	e	/	-	-	n	n	/	e	e	/	10	8	6	6	0		
7	/	-	-	-	e	e	/	/	-	-	e	/	n	n	/	/	-	/	/	-	-	-	/	-	-	-	e	/	-	10	14	4	2	0		
8	-	-	e	/	n	n	/	/	+	-	e	/	/	-	-	/	-	n	n	/	/	-	e	/	/	-	-	e	10	11	4	4	1			
9	/	/	-	e	/	-	-	e	e	/	/	-	e	/	-	-	-	e	/	n	n	/	/	-	-	e	/	-	-	10	11	6	3	0		
10	+	e	/	-	-	-	e	/	/	/	/	-	e	/	-	+	/	-	-	e	/	-	-	e	/	-	n	n	/	10	11	5	2	2		
11	e	/	-	-	-	/	-	e	/	-	n	n	/	/	/	+	-	e	/	-	-	-	n	n	/	/	-	e	/	10	11	4	4	1		
12	/	-	-	/	-	-	/	/	/	-	e	/	-	-	+	n	n	/	/	/	-	-	-	-	e	/	-	-	-	10	15	2	2	1		
13	-	/	n	n	/	/	+	-	/	/	/	-	-	-	e	e	e	/	-	-	e	e	e	/	/	/	-	-	-	10	11	6	2	1		
14	-	e	/	-	n	n	/	/	-	+	/	/	/	-	-	-	e	e	/	e	e	/	-	e	n	n	/	/	-	10	9	6	4	1		
15	-	-	e	/	/	-	e	e	/	-	e	e	/	-	-	/	n	n	/	/	-	-	-	/	-	-	-	n	n	9	11	6	4	0		
16	/	-	n	n	/	/	-	e	e	/	-	+	/	-	e	n	n	/	/	+	-	/	-	e	e	n	n	/	e	9	7	6	6	2		
17	e	/	-	e	e	/	-	-	n	n	/	/	+	-	e	/	-	-	e	/	/	-	n	n	/	/	-	e	/	10	9	6	4	1		
18	/	/	-	-	-	e	n	n	/	/	/	-	-	e	n	n	/	/	-	+	-	e	e	/	/	-	e	e	/	10	9	6	4	1		
19	-	/	-	-	-	/	+	-	e	/	n	n	/	/	-	+	e	/	-	-	-	/	/	/	/	-	-	-	10	14	2	2	2			
20	-	e	e	e	/	/	-	e	/	-	-	-	/	-	-	-	/	/	-	e	/	-	-	/	n	n	/	/	+	10	12	5	2	1		
21	n	n	/	/	-	-	e	/	n	n	/	/	-	-	e	/	-	-	e	/	-	-	e	e	e	/	/	-	+	9	10	6	4	1		
22	+	-	-	-	/	-	-	/	e	e	/	/	-	n	n	/	/	/	/	-	e	e	/	-	-	-	-	/	+	10	12	4	2	2		
23	-	-	/	-	e	/	n	n	/	/	-	-	-	e	/	/	-	e	/	/	/	-	-	-	-	e	/	-	+	10	13	4	2	1		
24	e	/	/	-	-	e	/	-	-	-	/	-	e	e	/	-	-	e	/	-	-	n	n	/	/	/	-	-	-	9	14	5	2	0		
25	n	n	/	/	-	-	/	-	-	/	-	e	n	n	/	/	-	-	-	-	n	n	/	/	-	-	/	e	e	e	9	11	4	6	0	
-	9	9	10	11	10	9	8	10	7	8	7	9	7	10	11	8	9	10	8	8	9	10	13	8	9	8	11	11	9							
e	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4							
n	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3							

その後、アメリカからはこの問題に関する論文が出版されなくなったが、アメリカでは保険制度の影響でスケジューリングの自由度も低く、ナース・スケジューリング研究の必要性がなくなったからだとも言われる。一方、ナース・スケジューリング問題の解決を望む国は多く、日本では1996年に新たなモデルが提案され(池上 他, 1996), 1998年からは日本のみならずイギリス、カナダからも一斉に論文(池上・丹羽, 1998; Dowsland, 1998; Jaumard et al., 1998; Millar and Kiragu, 1998; Nonobe and Ibaraki, 1998)が出始めた。多くの論文は、メタヒューリスティックアルゴリズムを評価するための1問題例として求解困難なこの問題を選んだものであったが、その理由にはメタヒューリスティックスの考え方が世に広まった直後だったことも考えられる。また、当時は、解いた問題例の具体的入力データを公開する論文がほとんど存在しなかった。Millar and Kiragu (1998)は、ネットワーク構造に基づく整数計画問題としてこの問題に取り組み、問題例データも公開していたが、対象とする問題サイズは非常に小さいもの(8 ナース・2週間分のスケジューリング)であった。現在では世界中の多くの研究者がナース・スケジューリング問題に取り組み(Burke et al., 2004), ベンチマークサイトも広まっている(University of Nottingham, 2006)。

ナース・スケジューリングは、何をもって最適な解(勤務表)とするかの定義が難しいだけでなく、例えば、定義できたとしても、解を得ること自体が難しい組合せ最適化問題として知られ

てきた。一方、近年の汎用ソルバーの高性能化から、目的関数や制約条件を規定できれば、厳密最適解を得ることも可能となり、最適解が知られていなかった問題例に対しても最適解を得ることに成功した結果も出てきた(乾・池上, 2010)。従って、今後の研究の興味は、アルゴリズムの構築だけでなく、評価尺度(目的関数)をどのように与え、最適解をどう評価するかや、解の修正の可能性を把握するために多様な解を得ることに向かうと考えられる。

本研究では、それらの方向性の基盤となるものとして、ナース・スケジューリングにおける実行可能解や良解の空間を把握する仕組みを提案する。解の空間を把握するために、ナース 1 人分の最適スケジュールを求める問題を部分問題として扱い、Ikegami and Niwa (2003) が提案した subproblem-centric approach における部分問題の解空間をネットワーク構造で表現することにより、部分問題の最適解を動的計画法で得ることや、良解の複数列挙を可能にする仕組みを提案し、解空間把握や勤務表作成の可能性について議論する。

具体的には、部分問題の全ての実行可能解を動的計画法の考え方に基づいてネットワーク構造で表す。ナース 1 人分の 1ヶ月の実行可能スケジュールは数 100 万存在すると考えられるが、それらすべてを列挙したり、比較評価しながら各ナースのスケジュールを組合わせていくことは、実行時間的に考えても現実的でない。そこで 1 ナースの実行可能スケジュールの全列挙と同等な解表現として、ネットワーク構造を利用し、ネットワーク上に設定したソースノードとシンクノードを結ぶすべての経路が実行可能スケジュールとなると同時に、すべての実行可能スケジュールがネットワーク上の経路として含まれるようにする方法を提案する。

本論文では、このネットワークの構築方法と、実際に構築した結果、そして、完成したネットワークを可視化して解の可能性を把握する仕組みについて報告する。以降、subproblem-centric approach のことを、部分問題軸アプローチ(池上, 2005)とよぶことにする。

## 2. ナース・スケジューリング

ナース・スケジューリングにおける制約条件は、各シフトの勤務メンバー構成に関わるシフト制約と、各ナースの勤務負荷に関わるナース制約とに、大きく分けることができる。

### シフト制約

毎日のシフトに必要な人数や適切なメンバーを確保するために、合計人数だけでなくスキルレベルや担当患者等で分けられたナースグループからの「各シフトにおける勤務人数」の上下限を設定する条件である。グループは、ナース集合の部分集合のことであり、1 人のナースが複数のグループに所属している(例えば、あるナースは担当患者により A チームに所属すると同時に、スキルレベル的にはベテラングループに所属するといったように、複数の部分集合の要素となる)ことが一般的である。

シフト制約は、例えば「夜勤には合計で 4 名、A チーム、B チーム、C チームからそれぞれ 1 名以上とし、新人同士の勤務を避け、ベテランが 1 名以上含まれるように考えている」場合、全員、A, B, C, 新人、ベテランの 6 種類のグループを設定し、毎日の夜勤に対して、勤務人数の(下限, 上限)をそれぞれ、(4,4), (1,2), (1,2), (1,2), (0,1), (1,4)といったように設定する。

### ナース制約

各ナースの社会的な生活や健康を守り、士気を保つためにも、休みや勤務の希望を考慮したり、無理な勤務にならないよう考慮する条件である。

具体的には、各ナースについて大きく以下の 3 つ、そして(3)についてはさらに詳しく 3 つを考える。

- (1) 各シフトに対する勤務回数や休み回数に上下限を設定する

- (2) セミナ等の確定している勤務, 休みや勤務の希望を達成する
- (3) 禁止するシフト並びを避ける
  - a) 同一シフトの連続回数の上下限を設定する
  - b) 同一シフトの勤務間隔日数の上下限を設定する
  - c) 異種シフトを含む禁止シフト並びを避ける

過去の勤務表を観察すると, 一般的に, シフト制約が重視され, ナースの休み希望を諦めるといったナース制約の緩和がみられるが, 長期的に看護の質を守るためには, ナースの健康や士気を守ることも非常に重要であり, シフト制約とナース制約の両方を満たすことが必要である.

これに関し, Ikegami and Niwa (2003) は, 各ナースの心身の健康状態を十分把握した人間でない限り「適切なナース制約の緩和」ができないことに着目した. さらに, シフト制約を全て守った解においては「シフト制約を満たしたままの修正が困難である」のに対し, ナース制約を全て守った解が「適切なナース制約の緩和を可能とする」と同時に「シフト制約をすべて満たした解への修正を容易にする」ことを病院現場で観察した. 従って, 現場に提供する暫定的な解は, ナース制約を全て守った下で, できる限りシフト制約を満たした解であるべきと主張し, 定式化としては, 暫定的な目的関数に「シフト制約を違反する度合いの最小化」を設定することを提案した. この提案に基づく定式化を以下に示す.

#### 意思決定変数

1 人のナースの実行可能スケジュールを「そのナースに関する全てのナース制約を満たすスケジュール」とし, ナース  $i$  が実行可能スケジュール  $p$  を割当てられるときに 1, そうでないときに 0 となる意思決定変数  $\lambda_{ip}$  を使う.

#### 記号

$M = \{1, 2, \dots, m\}$ : ナース番号の集合.

$N = \{1, 2, \dots, n\}$ : スケジューリング対象日の集合.

$W = \{0, 1, \dots, w\}$ : シフト番号の集合 (例, 0: 休み, 1: 日勤, 2: 準夜勤, 3: 深夜勤).

$R = \{r \mid r \text{ はグループ}\}$ : スキルや担当患者で分けられたグループの集合.

$G_r = \{i \mid \text{ナース } i \text{ はグループ } r \text{ に所属}\}, r \in R$ : グループ  $r$  に所属するナースに対応するナース番号の集合.

$a_{rjk}, r \in R, j \in N, k \in W$ :  $j$  日のシフト  $k$  に対するグループ  $r$  からの人数の下限值.

$b_{rjk}, r \in R, j \in N, k \in W$ :  $j$  日のシフト  $k$  に対するグループ  $r$  からの人数の上限値.

$P_i$ : ナース  $i$  の実行可能スケジュールの集合. ナース  $i$  の実行可能スケジュール.

$p \in P_i$  は  $\delta_{ipjk}$  (ナース  $i$  のスケジュール  $p$  において  $j$  日の勤務がシフト  $k$  なら 1, そうでなければ 0) を要素とするベクトル  $(\delta_{ip10}, \dots, \delta_{ipjk}, \dots, \delta_{ipnw})$  で表現される.

$\alpha_{rjk}^-, r \in R, j \in N, k \in W$ :  $j$  日のシフト  $k$  に対するグループ  $r$  からの人数が下限値  $a_{rjk}$  を下回る数を表す変数.

$\alpha_{rjk}^+, r \in R, j \in N, k \in W$ :  $j$  日のシフト  $k$  に対するグループ  $r$  からの人数が上限値  $b_{rjk}$  を上回る数を表す変数.

#### 定式化 1

$$(2.1) \quad \text{minimize} \quad \sum_{r \in R} \sum_{j \in N} \sum_{k \in W} (w_{rjk}^- \alpha_{rjk}^- + w_{rjk}^+ \alpha_{rjk}^+)$$

subject to

$$(2.2) \quad a_{rjk} - \alpha_{rjk}^- \leq \sum_{i \in G_r} \sum_{p \in P_i} \delta_{ipjk} \lambda_{ip} \leq b_{rjk} + \alpha_{rjk}^+, \quad r \in R, j \in N, k \in W$$

$$(2.3) \quad \sum_{p \in P_i} \lambda_{ip} = 1, \quad i \in M$$

$$(2.4) \quad \lambda_{ip} = 0 \text{ or } 1, \quad i \in M, p \in P_i$$

$$(2.5) \quad \alpha_{rjk}^-, \alpha_{rjk}^+ \geq 0, \quad r \in R, j \in N, k \in W$$

ここで、目的関数(2.1)式の  $w_{rjk}^-$  と  $w_{rjk}^+$  は、人数の過不足に対する重み付けであり、過不足数  $\alpha_{rjk}^-$ ,  $\alpha_{rjk}^+$  を(2.2)式で規定している。(2.3)式は、各ナースの各週にパターンをちょうど1つ選ぶ制約である。

各ナースの実行可能スケジュール集合  $P_i$  の要素数は数100万と考えられ、陽的に列挙したり保持することは現実的ではないことから、列生成の仕組みを持つアルゴリズムを意識した定式化といえる。

### 3. ナース・スケジューリングの部分問題

部分問題軸アプローチでは、1ナースの最適スケジュールを得る問題を部分問題としている。与えられた試行解(勤務表)に対し、他のナースのスケジュールを固定した下で、対象ナースのナース制約を満たしつつ「シフト制約を違反する度合い」を最小化する問題として定義している。そして、イテレーション毎にナース人数分の部分問題を解いて、目的関数値が最も小さくなった部分問題の解を次の試行解に採用することを、局所探索法やそれに基づく探索法(タブ探索など)の仕組みを使って繰り返し、全体の解を改善していく。

Ikegami and Niwa (2003) は、3交替制ナース・スケジューリング問題の部分問題を解くために列挙可能な程度の長さの実行可能部分スケジュール(例えば、長さ7日であれば高々1000種類)をあらかじめ用意し、分枝限定法を利用してそれらを組み合わせることにより、質の良い解を与えた。ナース25名の問題(Ikegami-3shift-DATA1)については、病院現場で作成された勤務表が39のシフト制約と71のナース制約を満たしてなかったことに対し、6つのシフト制約のみ満たさない勤務表を与えたと報告している。また、その後のMétivier et al. (2009)の論文が、同データにおいて63のシフト制約を満たしていないことを考えると、アルゴリズム的にも優れていることがわかる。

一方、Ikegami and Niwa (2003) では、この問題例を解くのに100時間(Pentium II, 266MHz)を要しており、実行時間が長くなるという問題を抱えると同時に、部分問題における実行可能解空間を把握するための情報も「実行可能部分スケジュール数」だけに留まっていた。

本論文でも、部分問題における実行可能部分スケジュール(以降、パターン)を基に解空間把握を試みるため、それに対応した新たな定式化を考える。パターンの長さは、隣接する期間のパターンが連結可能か否かでナース制約(3)を考慮できるよう7日とする。これは、一般にナース制約(3)が対象にする日数が7日(1週間)に収まるからである。

図1に、実行可能スケジュール例とそれを構成するパターンを示す。

例えば、深夜勤(n)の直後の準夜勤(e)や深夜勤直後2日間の日勤(-)が禁止されていた場合、1つめのパターンの直後に、3つめや4つめのパターンを連続して行うことは不可能である。

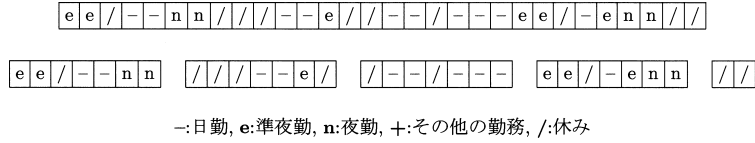
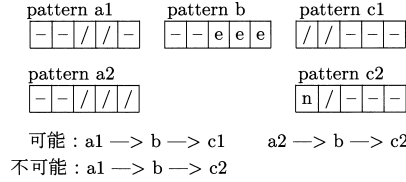


図 1. 実行可能スケジュール例とそれを構成する実行可能部分スケジュール (パターン).



a1 → b が可能かどうかは, その後のパターンに依存  
 b → c2 が可能かどうかは, その前のパターンに依存

図 2. 短すぎるパターンの例.

一方, 制約条件に比べてパターンの長さが短すぎると, 対象とする 2 つのパターンを同時に採用できるかは, さらに前後のパターンの情報がないと判定できない. 図 2 に示すパターンは, 短すぎて連続する 2 パターン間だけでは条件を満たすか判定できない例である.

1 週間に 1 回は休み (/) が入る (休みの間隔日数の上限が 6) といった条件があった場合, パターン a1 の後にパターン b を連続できるかどうかは, その後のパターンがどのようなものであるかによって異なる. 具体的には, パターン c1 であるなら可能だが, パターン c2 であるなら不可能である. 同様に, パターン b の後にパターン c2 を連続できるかどうかは, その前のパターンがどのようなものであるかによって異なる.

これに対し, ナース制約 (3) の対象日数の最大値を  $\hat{n}$  とすると, パターン長が  $\hat{n} - 1$  以上であれば図 2 に示した状況が発生しないため, 隣接期間のパターン間だけで判定できる (ただし, 後方に隣接期間のない最終期間に関しては  $\hat{n} - 1$  以上である必要はない).

以下に, 適正な長さのパターンを利用した場合の定式化を示す.

スケジュール期間内の週の数  $q$ ,  $h$  週に含まれる日の集合を  $N_h = \{j_1, \dots, j_{n_h}\}$ , ナース  $i$  の  $h$  週のパターンの集合を  $P_{ih}$  とし, 各パターン  $p \in P_{ih}$  を  $\delta_{ihpj_k}$  (ナース  $i$  の  $h$  週のパターン  $p$  において  $j$  日の勤務がシフト  $k$  なら 1, そうでなければ 0) を要素とするベクトル  $(\delta_{ihpj_1 0}, \dots, \delta_{ihpj_k}, \dots, \delta_{ihpj_{n_h} w})$  で表現する. また, 1 つのパターン  $p$  に対する各シフトの数を  $\rho_{ihpk} = \sum_{j \in N_h} \delta_{ihpj_k}$  とし, ナース  $i$  の 1 ヶ月におけるシフト  $k$  の勤務回数  $\rho_{ihpk}$  の下限と上限をそれぞれ  $c_{ik}, d_{ik}, k \in W$  とする. さらに, ナース  $i$  の  $h$  週のパターン  $p$  と翌週のパターン  $p'$  の間の連結可能性を  $\theta_{ihpp'}$  (可能なら 1, そうでなければ 0) で表すことにする. 意思決定変数としては, ナース  $i$  の  $h$  週のパターン  $p$  を採用するか否かを 1 と 0 で表す  $\lambda_{ihp}$  を使用する.

## 定式化 2

$$(3.1) \quad \text{Minimize} \quad \sum_{r \in G} \sum_{h=1}^q \sum_{j \in N_h} \sum_{k \in W} (w_{rjk}^- \alpha_{rjk}^- + w_{rjk}^+ \alpha_{rjk}^+)$$

subject to

$$(3.2) \quad a_{rjk} - \alpha_{rjk}^- \leq \sum_{i \in G_r} \sum_{p \in P_{ih}} \delta_{ihpjk} \lambda_{ihp} \leq b_{rjk} + \alpha_{rjk}^+,$$

$$r \in R, h = 1, \dots, q, j \in N_h, k \in W$$

$$(3.3) \quad c_{ik} \leq \sum_{h=1}^q \sum_{p \in P_{ih}} \rho_{ihpk} \lambda_{ihp} \leq d_{ik}, \quad i \in M, k \in W$$

$$(3.4) \quad \lambda_{ihp} + \lambda_{i \cdot h+1 \cdot p'} \leq 1 + \theta_{ihpp'}, \quad i \in M, h = 1, \dots, q-1, p \in P_{ih}, p' \in P_{i \cdot h+1}$$

$$(3.5) \quad \sum_{p \in P_{ih}} \lambda_{ihp} = 1, \quad i \in M, h = 1, \dots, q$$

$$(3.6) \quad \lambda_{ihp} = 0 \text{ or } 1, \quad i \in M, h = 1, \dots, q, p \in P_{ih}$$

$$(3.7) \quad \alpha_{rjk}^-, \alpha_{rjk}^+ \geq 0, \quad r \in R, j \in N, k \in W$$

(3.1), (3.2), (3.5)式は、それぞれ、定式化1の(2.1), (2.2), (2.3)式に対応する。(3.3)式と(3.4)式は、実行可能スケジュールを実行可能部分スケジュール(パターン)に分割したことで考慮できなくなったナース制約(1)と(3)にそれぞれ対応する。

定式化2における部分問題をシンプルに示すため、与えられた試行解でナース*i*以外のスケジュールを固定した場合のナース*i*の*h*週のパターン*p*のコスト  $f_{ihp}$  を以下のように求めておく。

$$f_{ihp} = \sum_{j \in N_h} \sum_{k \in W} \sum_{r \in G} \max \left\{ 0, w_{rjk}^- \left( a_{rjk} - g_{ir} \delta_{ihpjk} - \sum_{i' \in G_r, i' \neq i} \sum_{p' \in P_{i'h}} \delta_{i'h p' j k} \lambda_{i'h p'} \right), \right. \\ \left. w_{rjk}^+ \left( g_{ir} \delta_{ihpjk} + \sum_{i' \in G_r, i' \neq i} \sum_{p' \in P_{i'h}} \delta_{i'h p' j k} \lambda_{i'h p'} - b_{rjk} \right) \right\}$$

ここで、 $g_{ir}$  は、ナース*i*がグループ*r*のメンバーであれば1、そうでなければ0である値とする( $g_{ir} = |G_r \cap \{i\}|$ )。

部分問題*i* (ナース*i*) の定式化

$$(3.8) \quad \text{Minimize } \sum_{h=1}^q \sum_{p \in P_{ih}} f_{ihp} \lambda_{ihp}$$

subject to

$$(3.9) \quad c_{ik} \leq \sum_{h=1}^q \sum_{p \in P_{ih}} \rho_{ihpk} \lambda_{ihp} \leq d_{ik}, \quad k \in W$$

$$(3.10) \quad \lambda_{ihp} + \lambda_{i \cdot h+1 \cdot p'} \leq 1 + \theta_{ihpp'}, \quad h = 1, \dots, q-1, p \in P_{ih}, p' \in P_{i \cdot h+1}$$

$$(3.11) \quad \sum_{p \in P_{ih}} \lambda_{ihp} = 1, \quad h = 1, \dots, q$$

$$(3.12) \quad \lambda_{ihp} = 0 \text{ or } 1, \quad h = 1, \dots, q, p \in P_{ih}$$

本論文では、この部分問題における実行可能解(実行可能スケジュール)を把握することに焦点を絞る。

#### 4. 動的計画法に基づくネットワーク

前節の部分問題の定式化では、パターンの長さを 1 週間と設定することにより、すべての実行可能パターン列挙を容易にすると同時に、ナース制約(3)の「連続勤務回数」「勤務間隔日数」「禁止されるシフト並び」といった局所的だが扱い難い条件を、隣接するパターンを参照するだけで考慮できる利点がある。

1 週間分の実行可能パターンは、Ikegami and Niwa (2003) の 3 交替制勤務表作成データ (Ikegami-3shift-DATA1) においても 1 ナースの 1 週について数 100 (高々 1000) 程度である。それらのパターンが連結可能か否かも容易にチェックすることができ、 $\theta_{ihpp'}, h = 1, \dots, q-1, p \in P_{ih}, p' \in P_{i,h+1}$  をあらかじめマトリックス等に保持しておくことが可能である。

この問題を解くにあたって、2 つの方針が考えられる。1 つ目は、(3.9) 式を緩和した最短路問題を利用する方法であり、2 つ目は、部分問題を直接動的計画法で解く方法である。それぞれを 4.1 節と 4.2 節に示す。

##### 4.1 部分問題の緩和問題

各週のパターンをノードとし、 $h$  週のパターン  $p$  と  $h+1$  週のパターン  $p'$  に対し、 $\theta_{ihpp'} = 1$  であるならアークを設定する。ソースノード  $s$  とシンクノード  $t$  を加え、すべてのノードを含む集合を  $V$  とする。ソースノードから 1 週目のすべてのノード  $p \in P_{i1}$  の間にアークを設定し、 $q$  週目のすべてのノード  $p \in P_{iq}$  からシンクノードの間にアークを設定する。そして、これらすべてのアークの集合を  $A$  とする。さらに、 $h$  週のノード  $p$  に入るアークのコスト  $f_{p'p}, (p', p) \in A, p' \in V, p \in P_{ih}$  を  $f_{ihp}$  の値と同じにし、シンクノードの入るアークのコストを 0 ( $f_{pt} = 0$ ) とする。このように構成されたグラフ  $G = (V, A)$  上で最短路問題 (Dijkstra, 1959) を解くことを考える。

この最短路問題を解いて得られた解 (1ヶ月分のスケジュール) は、1ヶ月を通した「シフトや休みの回数に関する条件 (ナース制約(1))」以外の条件を満たした下での最適解である。よって、局所的には実行可能であっても、ナース制約(1)を満たしていない可能性があるので、この緩和解を直接利用するわけにはいかない。

そこで、この最短路問題に利用したネットワーク上でソースノードからシンクノードまでの経路を「上位から (3.9) 式を満たす解が得られるまで列挙すること」に対応する  $k$  最短路問題を解く。 $k$  最短路問題は、距離の短い方から上位  $k$  本の経路を探索する問題であり、MPS 法 (Martins et al., 1998) 等、高速なアルゴリズムも知られている。 $k$  最短路アルゴリズムの終了条件は「上位  $k$  番目までの保証ある良解が得られる」だが、この終了条件を「(3.9) 式を満たす解が得られる」に修正すればよい。

##### 4.2 動的計画法に基づくネットワークの構築

次に、部分問題を直接動的計画法で解くことを考える。本論文では、部分問題における良解の列挙も行うため、実行可能解を得るにも列挙を必要とする 4.1 節の方法より 1 経路を見つけるだけで実行可能解を得るこちらの方法に重点を置く。

まず、ナース  $i$  の  $h$  週のパターンが  $p$ 、 $h$  週までのシフト  $k \in W$  の累積回数が  $e_{ihk}$  に固定された下で「 $h$  週の子問題」を以下のように表してみる。



$h$  週のパターンと累積回数が固定された下での  $h$  週の子問題

$$(4.1) \quad g_h^*(p, e_{ihk}, k \in W) = \min \sum_{h'=1}^{h-1} \sum_{p' \in N_{h'}} f_{ih'p'} \lambda_{ih'p'}$$

subject to

$$(4.2) \quad \sum_{h'=1}^{h-1} \sum_{p' \in P_{ih'}} \rho_{ih'p'k} \lambda_{ih'p'} = e_{ihk} - \rho_{ihpk}, \quad k \in W$$

$$(4.3) \quad \lambda_{ih'p'} + \lambda_{i \cdot h'+1 \cdot p''} \leq 1 + \theta_{ih'p'p''}, \quad h' = 1, \dots, h-2, p' \in P_{ih'}, p'' \in P_{i \cdot h'+1}$$

$$(4.4) \quad \lambda_{i \cdot h-1 \cdot p'} \leq \theta_{i \cdot h-1 \cdot p'p}, \quad p' \in P_{i \cdot h-1}$$

$$(4.5) \quad \sum_{p' \in P_{ih'}} \lambda_{ih'p'} = 1, \quad h' = 1, \dots, h-1$$

$$(4.6) \quad \lambda_{ih'p'} = 0 \text{ or } 1, \quad h' = 1, \dots, h-1, p' \in P_{ih'}$$

従って,  $g_h^*(p, e_{ihk}, k \in W)$  は,  $1 < h \leq q$ ,  $p \in P_{ih}$ , 任意の  $e_{ihk}, k \in W$  に関して,

$$(4.7) \quad g_h^*(p, e_{ihk}, k \in W) = \min_{p' \in P_{i \cdot h-1}, \theta_{i \cdot h-1 \cdot p'p}=1} \{g_{h-1}^*(p', e_{ihk} - \rho_{ihpk}, k \in W) + f_{i \cdot h-1 \cdot p'}\}$$

$h=1$  の  $p \in P_{i1}$  ( $e_{i1k} = \rho_{i1pk}, k \in W$ ) に関して,

$$(4.8) \quad g_1^*(p, e_{i1k}, k \in W) = 0$$

と表現できる.

さらに, 累積回数  $e_{ihk}, k \in W$  のみ固定された  $h$  週までの問題の最適値は,

$$(4.9) \quad \min_{p \in P_{ih}} \{g_h^*(p, e_{ihk}, k \in W) + f_{ihp}\}$$

となり, 元の部分問題は, 以下のように表すことができる.

$$(4.10) \quad \min_{c_{ik} \leq e_{iqk} \leq d_{ik}, k \in W} \min_{p \in P_{iq}} \{g_q^*(p, e_{iqk}, k \in W) + f_{iqp}\}$$

この問題記述に従って, 動的計画法が利用できるネットワーク構築を考える. ネットワークを利用する理由は, 実行可能スケジュールの構成要素になりえない  $p \in P_{ih}$  と  $e_{ihk}, k \in W$  の値の組合せが存在することを意識し,  $g_h^*(p, e_{ihk}, k \in W)$  のための無駄な計算を回避したいからである.

そこで, ネットワークを構成するノードは, 各週  $h$  の「パターン  $p$  と累積回数  $e_{ihk}, k \in W$ 」(以降,  $\underline{p, e_{ihk}}$  と記述する) に対応させ, ソースノードとシンクノードを加える. ソースノードからは 1 週目のノード  $\underline{p, e_{i1k}}$  ( $e_{i1k} = \rho_{i1pk}, k \in W$ ) すべてにアークを設定する (1 週目の実行可能パターンは, 前月の勤務からの並びも考慮して実行可能なものとする). シンクノードへは,  $c_{ik} \leq e_{iqk} \leq d_{ik}, k \in W$  を満たす  $q$  週のノード  $\underline{p, e_{iqk}}$  からアークを設定する. さらに,  $h$  週のノード  $\underline{p, e_{ihk}}$  から  $h+1$  週のノード  $\underline{p', e_{i \cdot h+1 \cdot k}}$  へは,  $\theta_{ihpp'} = 1$ , かつ,  $e_{ihk} = e_{i \cdot h+1 \cdot k} - \rho_{i \cdot h+1 \cdot p'k}, k \in W$  が成り立つ場合のみアークを設定する.

従って, できあがったネットワーク上のソースノードからシンクノードまでのすべての経路が対象部分問題の実行可能解になるとともに, すべての実行可能解が経路としてこのネット

ワークに含まれることになる. さらに,  $p, e_{ihk}$  に入るアークに  $f_{ihp}$  をコストとして設定すれば, ソースノードからシンクノードまでの最短路は, 部分問題における最適解となり, 与えられた勤務表に対し(他のナースのスケジュールは固定した下で), シフトの勤務人数の過不足を最も少なくするような対象ナースのスケジュールを与えることになる.

$e_{ihk}, k \in W$  の値によっては, 前後の週にアークが設定されない場合があるので, ソースノードからの経路やシンクノードへの経路が存在しないノードやアークは対象としない工夫が必要である. 例えば,  $h$  週において, 以下の条件を満たす  $e_{ihk}$  を 1 つでも含んでいたら明らかに経路が存在しないノードである.

$$e_{ihk} < \sum_{h'=1}^h \min_{p \in P_{ih'}} \rho_{ih'pk}, \quad e_{ihk} > \sum_{h'=1}^h \max_{p \in P_{ih'}} \rho_{ih'pk}$$

$$e_{ihk} + \sum_{h'=h+1}^q \min_{p \in P_{ih'}} \rho_{ih'pk} > d_{ik}, \quad e_{ihk} + \sum_{h'=h+1}^q \max_{p \in P_{ih'}} \rho_{ih'pk} < c_{ik}$$

## 5. ネットワークの構築

(4.2) 節では, 定式化を利用してネットワークの構造を説明したが, ここでは, 構築方法についてさらに詳しく述べておく. その理由としては, ネットワークを構成するノード数やアーク数が膨大なため, 効率の悪い構築方法を採用すると非常に時間がかかってしまうからである. 例えば, ノードの実行可能性を判定して生成しながら構築する方法では, 翌週のノードを生成する際に無駄な列挙と判定が必要となり, 1 ナースのネットワークの構築だけでも(利用コンピュータにも依存するものの)数時間から数 10 時間を要し, 高速に部分問題の解を得たり, 解を把握するという目的と相反する. 対象月の勤務表作成において, 部分問題のネットワークは, ナース毎に 1 回構築してしまえば, 作り直すことなく何度でも利用できるが, できる限り高速に構築できることが望ましいと考え, 現時点では数 10 秒程度で構築することを目指す. そこで, 本研究では, シフトの回数だけに着目した圧縮ネットワークを初めに構築し, その後で, 実行可能パターンに対応するノードを持つネットワークに拡張することを考える.

具体的には,  $\rho_{ihpk}, k \in W$  と  $e_{ihk}, k \in W$  のそれぞれの値がすべて等しいノードセットを, 1 つのノード  $\hat{\rho}_{ihk}, e_{ihk}$  にまとめたネットワークを考え,  $h+1$  週のノード  $\hat{\rho}_{i \cdot h+1 \cdot k}, e_{i \cdot h+1 \cdot k}$  へは,  $e_{ihk} = e_{i \cdot h+1 \cdot k} - \hat{\rho}_{i \cdot h+1 \cdot k}$  が成り立つ場合のみアークを設定する. ソースノードからは 1 週目のすべてのノードにアークを設定し, シンクノードへは,  $q$  週の  $e_{iqk}$  が,  $c_{ik} \leq e_{iqk} \leq d_{ik}, k \in W$  となるノードからのみアークを設定する. こうすることにより, ソースノードからシンクノードへの経路は, 週を重ねるごとに各シフトに対する勤務累積数を増やしていく推移の可能性を表すと同時に, 実行可能な勤務累積数の推移のすべてがこのネットワークの経路として含まれることになる.

別の考え方をすると, 1 つのノード  $\hat{\rho}_{ihk}, e_{ihk}$  には,  $\rho_{ihpk} = \hat{\rho}_{ihk}, k \in W$  となる実行可能パターン  $p$  がすべて含まれているので, このノードに入るアークのコストを  $\min\{f_{ihp} | \rho_{ihpk} = \hat{\rho}_{ihk}, k \in W\}$  と設定すれば, このネットワーク上で最短路を求める問題は, 部分問題の定式化の (3.10) 式を緩和した問題を解くことになる. 本論文では, この圧縮ネットワークのことをシフト回数推移ネットワークとよぶことにする.

3 交替制勤務表作成データ (Ikegami-3shift-DATA1) を対象に, シフト回数推移ネットワークを構築した (Ikegami-3shift-DATA1 ではセミナの連続勤務数や間隔日数の制約について考慮する必要がなかったため, 本論文ではセミナを日勤扱いして考慮することにした). 25 名のナースにおいて, シフトの勤務回数の上下限が大きく異なる 2 名のナースを除けば, ほぼ同じ大

表 2. 3 交替データ (3shift-DATA1) の実行可能パターン (7 日) ネットワークのサイズ.

ナース番号	1	2	3	4	5	6	7	8	9	10
ノード数	62,574	12,479	41,949	3,317	9,407	18,620	50,181	24,734	15,281	38,478
アーク数	4,185,722	207,572	2,023,835	27,704	133,523	287,640	2,972,509	675,215	421,263	1,684,649
ナース番号	11	12	13	14	15	16	17	18	19	20
ノード数	30,963	3,234	3,347	5,215	28,062	16,930	7,634	8,978	3,689	11,167
アーク数	1,097,515	34,254	22,816	45,252	736,821	372,722	101,972	120,793	41,362	145,664
ナース番号	21	22	23	24	25					
ノード数	32,026	32,705	24,359	15,760	60,609					
アーク数	1,098,486	1,214,241	632,797	388,127	3,471,266					

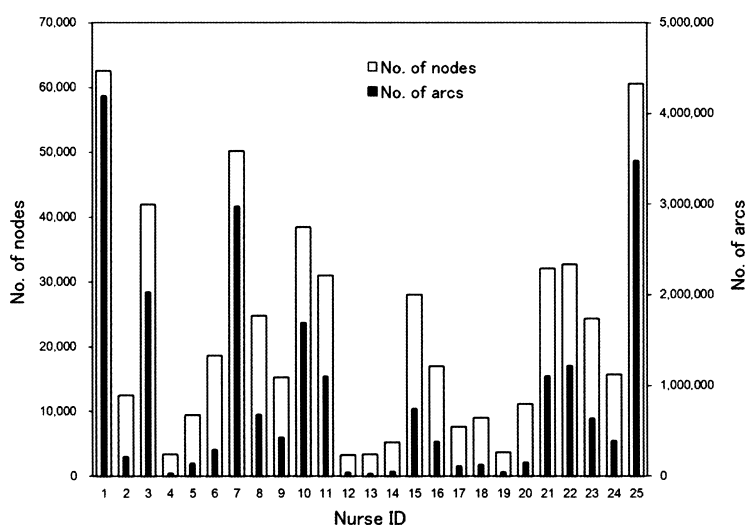


図 3. 3 交替データ (3shift-DATA1) の実行可能パターン (7 日) ネットワークのサイズ.

きさのシフト回数推移ネットワークが構築できた. 平均ノード数は 19,793 (最小 2,662, 最大 23,894), 平均アーク数は 327,088 (最小 16,966, 最大 406,846) であった. ノード数, アーク数ともに最小となったナースは深夜勤をちょうど 2 回勤務するという条件があり, 他のナースの勤務回数上下限の幅や数値が大きく設定されているのに比べて条件がタイトになっていた.

シフト回数推移ネットワークが構築されたら, 各ノードを対応する  $p, e_{ihk}$  に展開し, アークが設定されていた 2 つのノードがそれぞれ含む  $p, e_{ihk}$  と  $p', e_{i-h+1-k}$  が,  $\theta_{hpp'} = 1$  だったならば, 新しいネットワークにもアークを設定する. このようにノードを置き換える (展開する) ことにより, 目指していたネットワークを構築することができる. 完成したネットワークを実行可能パターンネットワークとよぶことにする.

前述のデータ (Ikegami-3shift-DATA1) においては, ナース毎に, 休み希望やセミナー等の予定が異なることから, 実行可能パターンネットワークのサイズも大きくならつきがあった. 表 2 に, それぞれのナースに対する部分問題の実行可能パターンネットワークのサイズ (ノード数とアーク数), 図 3 にそのグラフを示す. 平均ノード数は, 22,468 (最小 3,234, 最大 62,574), 平均アーク数は, 885,749 (最小 22,816, 最大 4,185,872) であった.

表 3. 3 交替データ (3shift-DATA1) の実行可能 (6 日) パターンネットワークのサイズ.

ナース番号	1	2	3	4	5	6	7	8	9	10
ノード数	36,605	6,523	25,815	2,048	4,942	10,581	28,757	17,292	8,842	27,618
アーク数	1,367,872	90,345	728,641	15,362	43,651	155,986	962,284	395,998	131,685	750,486
ナース番号	11	12	13	14	15	16	17	18	19	20
ノード数	22,807	2,684	1,885	4,485	14,899	12,146	6,959	3,976	2,789	4,828
アーク数	583,056	25,871	11,638	37,939	290,489	183,369	72,410	31,200	23,554	67,987
ナース番号	21	22	23	24	25					
ノード数	20,682	19,975	15,480	11,757	34,742					
アーク数	523,775	515,161	332,790	236,661	1,135,888					

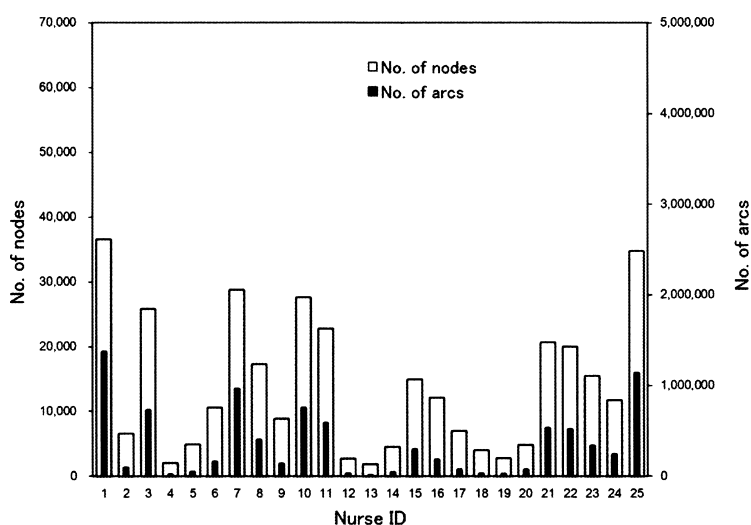


図 4. 3 交替データ (3shift-DATA1) の実行可能パターン (6 日) ネットワークのサイズ.

このデータでは、ナース制約 (3) が対象とする日数の最大が 7 日だったため、パターンの長さは 6 日でも可能である。表 3 と図 4 に示したパターン長を 6 にした場合のネットワークサイズを観察すると、そのサイズが小さくなっていることがわかる。これは、パターンの数が長さの階乗のオーダーで増えるからだと考えられるが、スケジューリング期間をいくつ ( $\lceil n/7 \rceil$  や  $\lceil n/6 \rceil$ ) に分けたかにも依存する。この問題例では、パターン長 7 でも 6 でも 5 区間に分かれるが、最終区間の長さがそれぞれ 2 と 6 となり、4 区間までのパターン数が非常に小さい場合には、最終区間のノード数の影響を受けてパターン長 6 の方がサイズが大きくなる場合もあると考えられる。

これまでの研究では、1 ナースの実行可能スケジュールを全列挙したり、その空間がどの程度の大きさであるか把握することが不可能だった。これに対し、提案する実行可能パターンネットワークは、各ナースの実行可能スケジュールを全列挙したのと同様な情報を持つことに成功できた。ネットワークのサイズの違いは、ソースノードからシンクノードへの経路の数の違い、つまり実行可能スケジュール数の違いを正確に示すわけではないが、大きな依存関係は存在すると考えられる。よって、各ナースのスケジュールの自由度を大まかに捉えることができる。

ナース毎にネットワークのサイズが異なる理由は、ナース制約のパラメータの違いによる。例えば、ナース 12 のネットワークサイズが非常に小さい理由は前にも簡単に述べたが、多くのナースが準夜勤を 4~6 回、深夜勤を 2~4 回もしくは 3~6 回勤務できるのに対し、準夜勤、深夜勤ともちょうど 2 回勤務するというタイトな設定になっているからだと考えられる。また、休み希望や固定勤務が 5 日以上設定されているナース 2, 4, 13, 14, 19 のネットワークサイズは比較的小さく、逆に、休み希望や固定勤務が全く設定されていないナース 1, 25 のネットワークサイズは大きい傾向がみられる。

問題例にも依存するものの、1 つのネットワークの構築には、20 秒~30 秒程度を要し (Intel (R) Xeon (R) CPU E5335 Quad Core @2.00GHz, Memory 24G)、構築アルゴリズムのデータ構造や実装方法にさらなる工夫が必要だが、これまで知りえなかった情報を生み出すという意味では、現時点では十分な速さと考える。

ここで提案した構築方法が効率よい理由としては、各週に列挙した高々 1000 程度の実行可能パターンと、隣接する週のパターンとの連結可能性を初めに求めておくことにより、シフト回数推移ネットワークが構築できた時点で、実質的には目指すネットワーク (実行可能パターンネットワーク) ができているからだと考える。なお、2 交替制のデータ (Ikegami-2shift-DATA1 を緩和した問題: 土日連休を 1 回確保する条件を緩和; ナース数 28) についても同様なネットワーク構築に成功している。

ここで利用した 3 交替制データ (Ikegami-3shift-DATA1) は、Ikegami and Niwa (2003) やベンチマークサイト (University of Nottingham, 2006) で紹介され、多くの研究者に取り組みられてきた問題例である (Métivier et al., 2009)。

## 6. ネットワークの利用

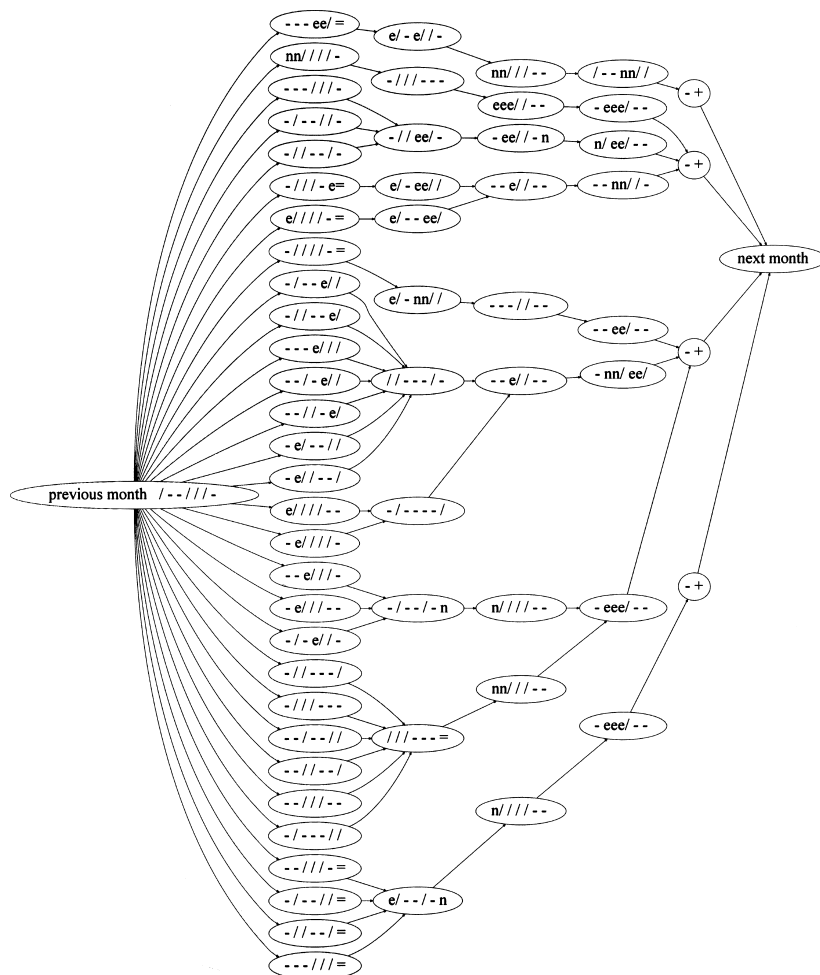
実行可能パターンネットワークを構築したことで、個々のナースの実行可能スケジュールの空間を把握しやすくなり、各ナースに対する条件の緩和や追加による、実行可能スケジュール集合の大きさや特徴の変化も把握できるようになった。

また、前述したように、実行可能パターンネットワークのアーク (ノード  $p, e_{ihk}$  に入るアーク) に  $h$  週にパターン  $p$  を採用した場合の各シフトの人数過不足の度合いである  $f_{ihp}$  をコストとして設定すれば、ソースノードからシンクノードへの最短路は、部分問題における最適解に対応する。構築したネットワークにおいて、ある試行解 (あるコスト設定) に対して最短路を得るための実行時間の平均は、3 交替制データの 7 日パターンのネットワークでは 0.065 秒、6 日パターンのネットワークでは 0.046 秒であった (Intel (R) Xeon (R) CPU E5335 Quad Core @2.00GHz, Memory 24G)。この時間もデータ構造等の工夫により短縮が可能だと考える。

部分問題の最適解が最短路問題を解くことによって高速に得られれば、Ikegami and Niwa (2003) のアルゴリズムの「部分問題を解く分枝限定法」をこれに置き換えることにより、全体の勤務表作成も可能になる。

一方、実行可能パターンネットワークのノード数とアーク数は膨大であり、実際にネットワークを表示して視覚的に把握することは難しい。しかし、本来、我々の興味は、その中で現在の勤務表を改善できる可能性のある優れたスケジュール群であると考えられる。そこで、実行可能パターンネットワークに含まれる良解のみを抽出し、それを構成するノードやアークを表示して、良解空間を提供することを考える。具体的には、実行可能パターンネットワーク上で  $k$  最短路を求め、最適スケジュールを 1 つ得るだけでなく複数の良解を得ることにより、勤務表の改善案を提案することや、各ナースにおける勤務変更の可能性把握を支援する。

実行可能パターンネットワーク上の  $k$  最短路 ( $k=30$ ) を求め、そこで採用されたノードとアー

図 5.  $k$  最短路を構成するノードとアーク.

クで構成されるネットワークのイメージを図 5 に示す(ここでは, 問題例のナース 20 のネットワークを対象に, 便宜的に  $f_{ihp}$  の値をランダムに与えてコストとして設定した場合の結果を示した).

ソースノードが前月末の 7 日分のスケジュールを表し, シンクノードが翌月のスケジュールを表す. その他のノードは, 左側より 1 週目から 5 週目までの累積シフト回数情報を伴ったパターン(回数情報は非表示)に対応する. ここでは, 見やすさのため  $k=30$  としたが,  $k=50, 100$  でも高速にネットワークを得ることができる.

さらに, このように良解を含んだ縮小したネットワークを有効に利用すれば, 探索空間は狭まり高速なスケジューリングも可能になると考える.

## 7. おわりに

本論文では、これまで実行可能解の空間の把握が困難だった問題に対し、部分問題の実行可能解のすべてをネットワーク表現することにより、全列挙したのと同様な解の把握を可能とするだけでなく、他のナースに対するスケジュールを固定した下での最適スケジュールや上位の好ましいスケジュールを高速に得る仕組みを構築した。また、このネットワークを利用して部分問題を解くことによる勤務表作成の概要を示した。

最適化技術においては、一般に、目的関数を一意に設定することが求められるが、ナース・スケジューリングをはじめとする、人間の活動が大きくシステムに関わっている問題においては、潜在的な評価尺度が複数存在する場合も多く、目的関数の設定は困難な課題となる。また、数値化しにくい人間の満足度やバランス感覚の表現も数値演算を基本とする最適化技術では扱いにくい対象である。暫定的に設定した目的関数に対する最適解も、現実利用されるためには、明示されていなかった制約や評価尺度に対して何らかの調整(修正)が必要となる。

一方、人間にとっての最適解は1つでないと同時に、解を得るためのプロセスが重要であり、自らが作った最適解であるという納得感が存在しない限り、現場で実際に利用されることはない(田辺 他, 2009)。納得感を実現するためには、最適化アルゴリズムが与えた最適解と、人間にとっては最適であり得たかもしれない解との関係をなんらかの形で提供する必要がある。ある条件の下、最適解とほぼ等しい評価値を持つ解が膨大に存在する場合もあれば、非常に特異な解である場合、さらには、その特異さが現実的には非常に好ましい場合もあれば、不自然な場合もあり得る。

意思決定における納得感を得るためには、選択された解の位置づけ(意味)の把握が重要であり、他に選択できる可能性やその意味を十分に理解し、修正の試行錯誤も含め、問題の解空間全体を把握するというプロセスが必要である。

本研究では、このプロセスを支援する数理的な仕組みの1つとして、ネットワーク構造を利用し、実行可能解空間を表現することに取り組み、その人間にとっての最適解を選択できる可能性を広げた。できあがったネットワークを人間にとって視覚的にも把握しやすいよう、意味のある部分に絞り込んで提供するなど、良解の分布に対する情報提供を可能にし、これまでの意思決定において、気づかなかった制約や評価尺度を意識できるようにするといった、人間の思考に調和する最適化、人間の暗黙知に対してロバスト性を持つ最適化の研究につながると考える。

## 謝 辞

本論文に貴重なご助言を頂いた査読委員の先生方に心より感謝致します。本研究について議論を頂いた株式会社数理システムの田辺隆人氏、成蹊大学院生の山口淳史さん、井村亮介さんに心より感謝致します。本研究の一部は、文部科学省私立大学戦略的基盤形成支援事業研究費の成果、池上の統計数理研究所客員教授としての成果、政策研究大学院大学客員研究員としての成果を含んでいます。

## 参 考 文 献

- Arther, J. L. and Ravindran, A. (1981). A multiple objective nurse scheduling model, *AIIE Transactions*, **13**, 55–60.
- Bell, P. C., Hay, G. and Liang, Y. (1986). A visual interactive decision support system for workforce

- (nurse) scheduling, *INFOR*, **24**, 134–145.
- Burke, E. K., De Causmaecker, P., Berghe, G. V. and Landeghem, H. V. (2004). The state of the art of nurse rostering, *Journal of Scheduling*, **7**, 441–499.
- Dijkstra, E. W. (1959). A note on two problems in connection with graphs, *Numeriche Mathematics*, **1**, 269–271.
- Dowland, K. A. (1998). Nurse scheduling with tabu search and strategic oscillation, *European Journal of Operational Research*, **106**, 393–407.
- 池上敦子 (2005). ナース・スケジューリング — 調査・モデリング・アルゴリズム —, *統計数理*, **53**, 231–259.
- 池上敦子, 丹羽明 (1998). ナース・スケジューリングに有効なアプローチ: 2 交替制アルゴリズムにおける実現, *Journal of the Operations Research Society of Japan*, **41**, 572–586.
- Ikegami, A. and Niwa, A. (2003). A subproblem-centric model and approach to the nurse scheduling problem, *Mathematical Programming*, **97**, 517–541.
- 池上敦子, 丹羽明, 大倉元宏 (1996). 我が国におけるナース・スケジューリング問題, *オペレーションズ・リサーチ*, **41**, 436–442.
- 乾伸雄, 池上敦子 (2010). ナーススケジューリング問題における混合整数線形計画問題と充足性判定問題による厳密解法の比較, *オペレーションズ・リサーチ*, **55**, 706–712.
- Jaumard, B., Semet, F. and Vovor, T. (1998). A generalized linear programming model for nurse scheduling, *European Journal of Operational Research*, **107**, 1–18.
- Martins, E. Q. V., Pascoal, M. M. B. and Santos, J. L. E. (1998). The  $k$  shortest paths problem, Research Report, The University of Coimbra, Portugal.
- Métivier, J., Boizumault, P. and Loudni, S. (2009). Solving nurse rostering problems using soft global constraints, *Lecture Notes in Computer Science (Principles and Practice of Constraint Programming)*, **5732**, 73–87.
- Millar, H. H. and Kiragu, M. (1998). Cyclic and non-cyclic scheduling of 12 h shift nurses by network programming, *European Journal of Operational Research*, **104**, 582–592.
- Miller, H. E., Pierskalla, W. P. and Rath, G. J. (1976). Nurse scheduling using mathematical programming, *Operations Research*, **24**, 857–870.
- Nonobe, K. and Ibaraki, T. (1998). A tabu search approach to the constraint satisfaction problem as a general problem solver, *European Journal of Operational Research*, **106**, 599–623.
- 田辺隆人, 岩永二郎, 多田明功, 池上敦子 (2009). 「納得」を生み出すスケジューリングアルゴリズムとソフトウェア制約充足を超えて: 実行可能領域の直観的把握, *スケジューリング・シンポジウム 2009*, 169–173, 岡山.
- University of Nottingham (2006). Personnel Scheduling Data Sets and Bench-marks, <http://www.cs.nott.ac.uk/tec/NRP/>.
- Warner, D. M. (1976). Scheduling nursing personnel according to nursing preference: A mathematical programming approach, *Operations Research*, **24**, 842–856.



## A Network Representation of Feasible Solution Space for a Subproblem of Nurse Scheduling

Hiroki Akita and Atsuko Ikegami

Faculty of Science and Technology, Seikei University

This paper deals with the nurse scheduling problem. When making a roster for nurses in a hospital, a scheduler has to assign the required number of nurses with specified skill levels to each shift while balancing the workload among the nurses. This problem is known to be difficult to solve using computers, because it is impossible to explicitly describe all of considerations. This paper describes a system for enumerating good solutions and a tool for clarifying the space of feasible solutions in order to help the scheduler's decision. First we focus on a subproblem to find the optimal schedule or feasible schedules for a nurse. We then construct a network in which any path from the source node to the sink node represents a feasible schedule and all the schedules are included. Given that the other nurses' schedules are fixed, we can obtain the optimal and other good schedules for each nurse efficiently by finding the shortest path and the  $k$ -shortest paths on the corresponding network. We also show how to construct the network for a nurse and features of the networks generated for all nurses. Though further studies are needed to reduce the size of the networks in terms of the numbers of nodes and arcs, it will be made clear that they can be of tremendous help in scheduling nurses efficiently.